JPL D-26304

Software Cost Analysis Tool User Document

Version 0

Prepared by: **Karen Lum**
Erik Monson

June 11, 2003

Jet Propulsion Laboratory
Pasadena, California

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 What is SCAT?

SCAT is an acronym for Software Cost Analysis Tool. It is an Excel-based version of the University of Southern California's Constructive Cost Model II (COCOMO II).[1] COCOMO II is a software cost model developed by Barry Boehm, et al. COCOMO II has been validated using JPL data and provides reasonable flight and ground software estimates in the JPL environment.[2] It is our strong recommendation that you determine the viability of the uncalibrated COCOMO II model within your organizational environment. It is not in the scope of this document to discuss local calibrations, nor is this tool able to perform local calibrations.

An adaptation to COCOMO II is that SCAT allows the use of a range of inputs (Low, Most Likely, High) to capture uncertainty and generate probability distributions through the use of Monte Carlo techniques. Therefore, unlike the USC version, SCAT produces distributional estimates.

1.2 Purpose and Features

The purpose of this tool is to provide software cost estimates using a Monte Carlo simulation and COCOMO II cost drivers and algorithms.

SCAT uses the most recent COCOMO II.2000 algorithms, published in *Software Cost Estimation with COCOMO II.*

For More Help Contact:

Karen Lum
(818) 354-5036 (Office)
karen.t.lum@jpl.nasa.gov

Erik Monson
(818) 393-7478 (Office)
erik.w.monson@jpl.nasa.gov

[1] Boehm, B., et al., *Software Cost Estimation with COCOMO II*, Upper Saddle River, New Jersey, Prentice Hall PTR: 2000.

[2] Lum, K., Powell, J., and Hihn, J. "Validation of Spacecraft Software Cost Estimation Models for Flight and Ground Systems," International Society of Parametric Analysts 2002 Conference Proceedings, May 2002.

2.0 GETTING STARTED

2.1 System Requirements

Currently, SCAT is only supported for PCs with Microsoft Excel. The Monte Carlo add-in may be unstable in the Mac environment and therefore is not supported.

2.2 Installing SCAT files

SCAT is composed of a set of the following linked files with additional helper/add-in files:

File	Details
SCAT.xls	User Workbook. Estimating done here.
mctool_xl2k.xla	Monte Carlo Excel-add in. Contains VB code needed to enable the Monte Carlo simulations.
mctool.hlp	Monte Carlo help file

Step **Action**

1) Create a new SCAT folder on your desktop or any other convenient place on your hard drive.
2) Download the SCAT workbook into your SCAT folder.
3) Put mctool_xl2k.xla file in C:\Program Files\Microsoft Office\Office\Library
4) Put mctool.hlp file in C:\Program Files\Microsoft Office\Office\1033
5) To start, double click to open the SCAT.xls workbook file. This is the only file you need to open.

This file can be copied as many times as needed to hold and analyze the cost estimates. It is convenient to save the copies or revision in an archive subfolder within your SCAT folder.

3.0 SCAT CONTROLS AND BUTTONS

SCAT's main controls are located at the bottom of each input sheet. (Figure 1)

3.1 Add New Module

When this button is clicked, an input sheet is added. You will need an input sheet for **each software element** or piece that you wish to estimate. It is also acceptable to estimate the software project as a whole. In this case, you would not need to add any extra modules. However, it is recommended that you break your software project down into smaller pieces or functions (For JPL personnel, see Software Cost Estimation Handbook, D-24385, JPL Rules! DocID 62332; others, please see Handbook for Software Cost Estimation, D-26303). The minimum total size that the tool is calibrated for is 2,000 lines of code.

If you should inadvertently create too many input sheets, **do not attempt to delete the sheets**. Simply set all size inputs to '0' and the inputs on this sheet will have no effect.

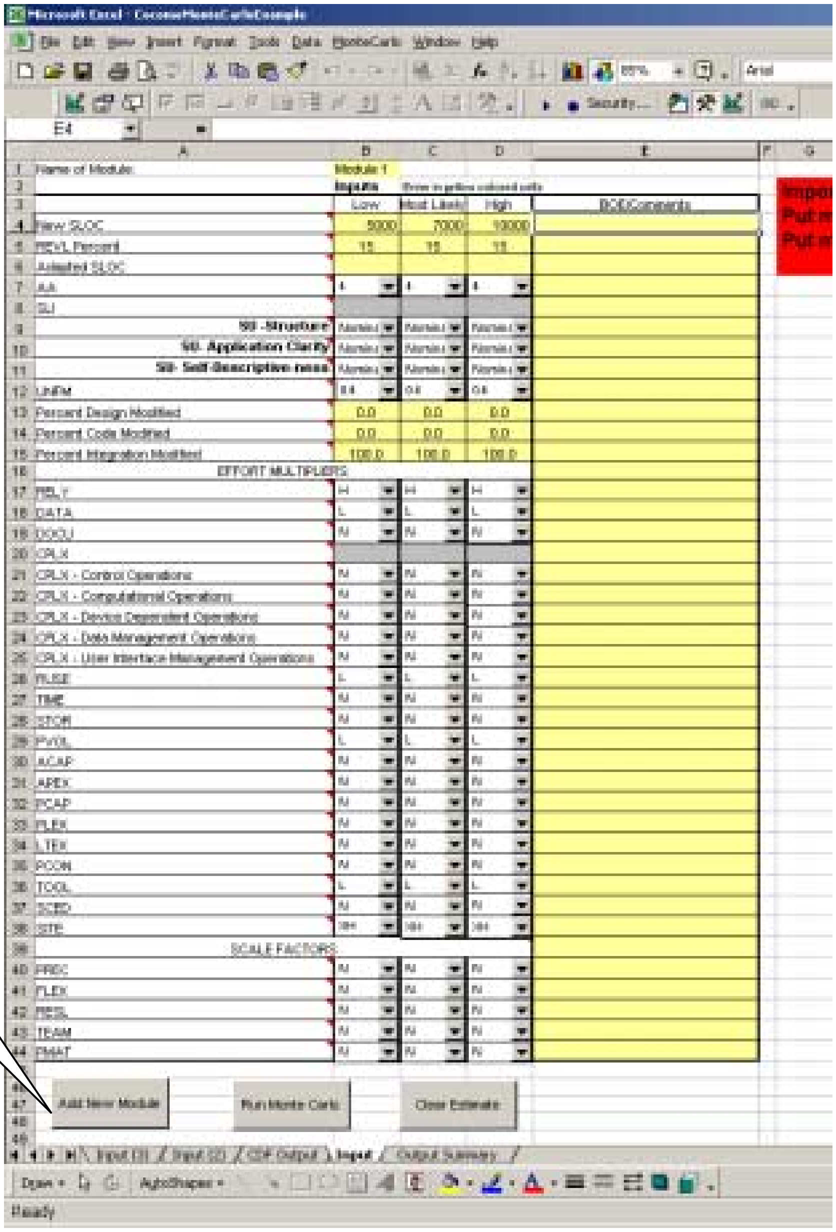
3.2 Run Monte Carlo

This button runs a Monte Carlo simulation in order to generate an estimate. Click this button once you are done entering and selecting inputs for all of your software pieces.

3.3 Clear Estimate

The 'Clear Estimate' feature should be used with caution. Clicking this button will delete **all sheets** with the exception of one of your input sheets and begin a new estimate session. Only click this button after you have saved all previous work, and if you are positive you wish to start a new estimate. After you click this button, a warning message will appear. If you are sure you wish to begin a new estimate from scratch, select "OK." If you do not wish to lose any of your current input sheets, select "No."

It is not necessary to clear your estimate to analyze various estimate scenarios. See "Changing Your Inputs to Analyze the Tradeoffs" in the next chapter.


Microsoft Excel - CocomoMonteCarloExample
File Edit View Insert Format Tools Data MonteCarlo Window Help
85%
Arial
Security...
E4
Add New Module
Run Monte Carlo
Clear Estimate
Input (1) / Input (2) / CDF Output / Input / Output Summary
Draw AutoShapes
Ready

	A	B	C	D	E
1	Name of Module:	Module 1			
2		Inputs	Enter in yellow colored cells		
3		Low	Most Likely	High	BOE/Comments
4	New SLOC	5000	7000	10000	
5	REVL Percent	15	15	15	
6	Adapted SLOC				
7	AA	1	1	1	
8	SU				
9	SU -Structure	Nomin	Nomin	Nomin	
10	SU- Application Clarity	Nomin	Nomin	Nomin	
11	SU- Self-descriptive ness	Nomin	Nomin	Nomin	
12	UNFM	0.4	0.4	0.4	
13	Percent Design Modified	0.0	0.0	0.0	
14	Percent Code Modified	0.0	0.0	0.0	
15	Percent Integration Modified	100.0	100.0	100.0	
16	EFFORT MULTIPLIERS				
17	RELY	H	H	H	
18	DATA	L	L	L	
19	DOCU	N	N	N	
20	CPLX				
21	CPLX - Control Operations	N	N	N	
22	CPLX - Computational Operations	N	N	N	
23	CPLX - Device Dependent Operations	N	N	N	
24	CPLX - Data Management Operations	N	N	N	
25	CPLX - User Interface Management Operations	N	N	N	
26	RUSE	L	L	L	
27	TIME	N	N	N	
28	STOR	N	N	N	
29	PVOL	L	L	L	
30	ACAP	N	N	N	
31	APEX	N	N	N	
32	PCAP	N	N	N	
33	PLEX	N	N	N	
34	LTEX	N	N	N	
35	PCON	N	N	N	
36	TOOL	L	L	L	
37	SCED	N	N	N	
38	SITE	VH	VH	VH	
39	SCALE FACTORS				
40	PREC	N	N	N	
41	FLEX	N	N	N	
42	RESL	N	N	N	
43	TEAM	N	N	N	
44	PMAT	N	N	N	

Controls and Buttons are at the bottom of the input sheet.

Figure 1. Software Controls/Buttons are on the Input Sheets

3.4 Other Controls

There are some buttons on the "CDF Output" sheet that allow you to unlock the sheet for editing and relock that sheet. See "Modifying the CDF Chart" or "Copying the CDF Chart to Another Media" in the "Advanced Features" Section for information on how to use these controls.

4.0 BUILDING A SOFTWARE ESTIMATE

You can estimate a software project as a whole, or subdivide it into smaller elements, such as by modules, CSCI's or functionality. Since COCOMO II is not calibrated for projects below 2,000 SLOC in total size, the size of each module should be no less than 2,000 SLOC. The maximum size input for SLOC is 9,999,999.

4.1 Input Area

All input sheets are named "Input" followed by a number in parenthesis. An input sheet (Figure 2) needs to be created for each software element you wish to estimate by clicking on the "Add New Module" button at the bottom of the input sheet.

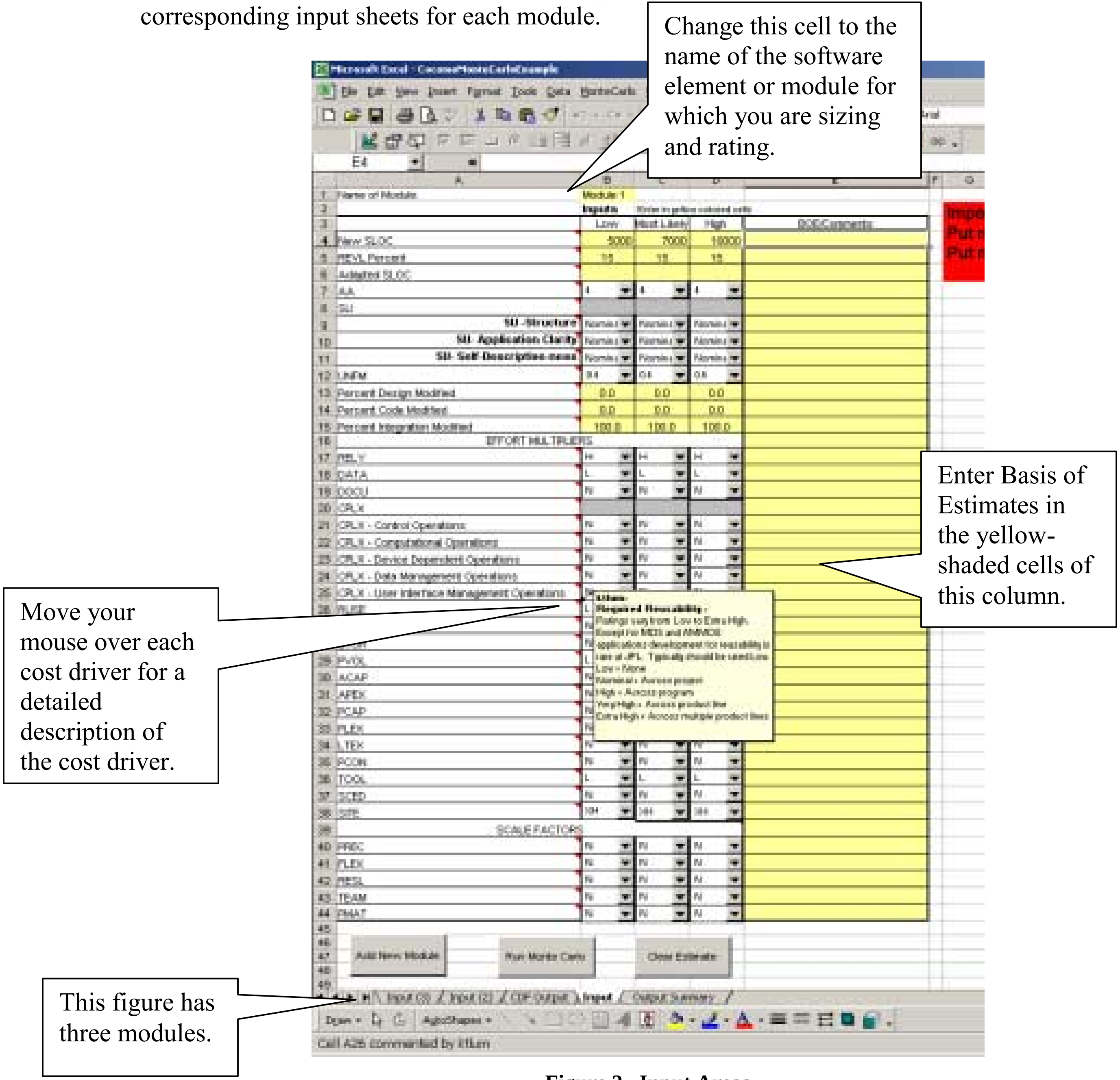
Each input sheet consists of a table with drop down boxes and yellow-shaded cells. The yellow-shaded cell at the top of the input sheet is for entering the name of the software element or module to be estimated. The drop down boxes have ratings that correspond to COCOMO II cost drivers. Select the rating(s) for each cost driver that best describes the software element you are estimating. Each of the parameters can be rated on a scale that generally varies from "very low" to "extra high"; some parameters do not use the full scale. Each rating has a corresponding real number based upon the factor and the degree to which the factor can influence productivity. For the Complexity cost drivers, there is also a choice of "N/A" (Not Applicable) for the complexity areas that do not apply to your type of software. If a complexity area is not applicable, **be sure to set all three ratings to "N/A" for "low," "most likely," and "high."**

If you are reasonably certain of the accuracy of a rating, select the same rating for the low, most likely, and high. If you are not sure, you may select three different ratings. However, make sure that the ratings increase in scale. For example, your "low" input must be less than or equal to your "most likely" input, and your "most likely input" should be less than or equal to the "high" input: Nom, Nom+25, High or Low, Low, Low+75 are also acceptable inputs. However, if you enter a descending set such as: High, Nom, Nom+25, you will get a warning message to fix this error.

All yellow-shaded cells are for user input. Enter the number of **new** logical lines of code in the first yellow shaded row. If there is inheritance or reuse, enter the number of lines of code to be inherited or reused in the row labeled "Adapted SLOC." Then enter the amount of design modification, code modification, and retesting required in the yellow-shaded rows following. Percentages should be entered as whole numbers. For example, REVL asks you to enter the percentage of code you think will be thrown away due to requirements volatility. If you believe that 15% of your code will need to be thrown away due to requirements volatility, enter '15'. Bases of estimates and notes can be entered in the yellow-shaded cells of the fifth column.

There are comment text boxes that appear as you scroll your mouse over each cost driver to aid you in selecting your ratings. Appendix A also lists many of the COCOMO II parameters and their associated rating scales. If you are a JPL employee and require additional assistance, contact a software cost expert from the Software Quality Improvement (SQI) Project (http://software/), otherwise you may choose to reference *Software Cost Estimation with COCOMO II.*[3]

Select and enter information for each subsequent module in a similar manner into the corresponding input sheets for each module.


Microsoft Excel - CocomoMonteCarloExample
File Edit View Insert Format Tools Data MonteCarlo
E4
Add New Module
Run Monte Carlo
Clear Estimate
Input (3) / Input (2) / CDF Output / Input / Output Summary
Draw AutoShapes
Cell A26 commented by ittum
Impo
Put n
Put n

	A	B	C	D	E
1	Name of Module	Module 1			
2		Inputs	Enter in yellow colored cells		
3		Low	Most Likely	High	BOE/Comments
4	New SLOC	5000	7000	10000	
5	REVL Percent	15	15	15	
6	Adapted SLOC				
7	AA	4	4	4	
8	SU				
9	SU -Structure	Nomina	Nomina	Nomina	
10	SU- Application Clarity	Nomina	Nomina	Nomina	
11	SU- Self-Descriptive-ness	Nomina	Nomina	Nomina	
12	UNFM	0.4	0.4	0.4	
13	Percent Design Modified	0.0	0.0	0.0	
14	Percent Code Modified	0.0	0.0	0.0	
15	Percent Integration Modified	100.0	100.0	100.0	
16	EFFORT MULTIPLIERS				
17	RELY	H	H	H	
18	DATA	L	L	L	
19	DOCU	N	N	N	
20	CPLX				
21	CPLX - Control Operations	N	N	N	
22	CPLX - Computational Operations	N	N	N	
23	CPLX - Device Dependent Operations	N	N	N	
24	CPLX - Data Management Operations	N	N	N	
25	CPLX - User Interface Management Operations	N	[illegible]	[illegible]	
26	RUSE	L	[illegible]	[illegible]	
27	[illegible]	N	[illegible]	[illegible]	
28	[illegible]	N	[illegible]	[illegible]	
29	PVOL	L	[illegible]	[illegible]	
30	ACAP	N	[illegible]	[illegible]	
31	APEX	N	[illegible]	[illegible]	
32	PCAP	N	[illegible]	[illegible]	
33	PLEX	N	[illegible]	[illegible]	
34	LTEX	N	N	N	
35	PCON	N	N	N	
36	TOOL	L	L	L	
37	SCED	N	N	N	
38	SITE	VH	VH	VH	
39	SCALE FACTORS				
40	PREC	N	N	N	
41	FLEX	N	N	N	
42	RESL	N	N	N	
43	TEAM	N	N	N	
44	PMAT	N	N	N	

ittum:
Required Reusability:
Ratings vary from Low to Extra High.
Except for MDS and AMMOS applications development for reusability is rare at JPL. Typically should be used Low.
Low = None
Nominal = Across project
High = Across program
Very High = Across product line
Extra High = Across multiple product lines

Change this cell to the name of the software element or module for which you are sizing and rating.

Enter Basis of Estimates in the yellow-shaded cells of this column.

Move your mouse over each cost driver for a detailed description of the cost driver.

This figure has three modules.

Figure 2. Input Areas

[3] Boehm, B., et al., *Software Cost Estimation with COCOMO II*, Upper Saddle River, New Jersey, Prentice Hall PTR: 2000.

4.2 Output Areas

A summary sheet named “Output Summary” (Figure 4) lists the mean output values that the USC-based version of COCOMO II would output for the most likely case.

The sheet named “CDF Output” (Figure 3) is the main output after a Monte Carlo simulation is run. The Cumulative Distribution Function (CDF) is a curve produced from the random draws to aid you in understanding the probability of the estimate.

4.2.1 CDF Output Sheet

Two CDF charts are available on the “CDF Output” sheet: (1) Effort CDF and (2) Cost CDF. (Figure 3)

Reading the Effort CDF Chart

The x-axis is effort in work-months; the y-axis is probability or likelihood of occurrence. *Options and features*: You may view your CDF charts with or without the lines. Use the check boxes to turn the lines on or off. The following lines can be viewable by selecting the check boxes:

- Plus or minus one Standard deviation
- 70th percentile is the recommended budget with reserves
- 50th percentile is the recommended minimum budget without reserves

Reading the Cost CDF Chart

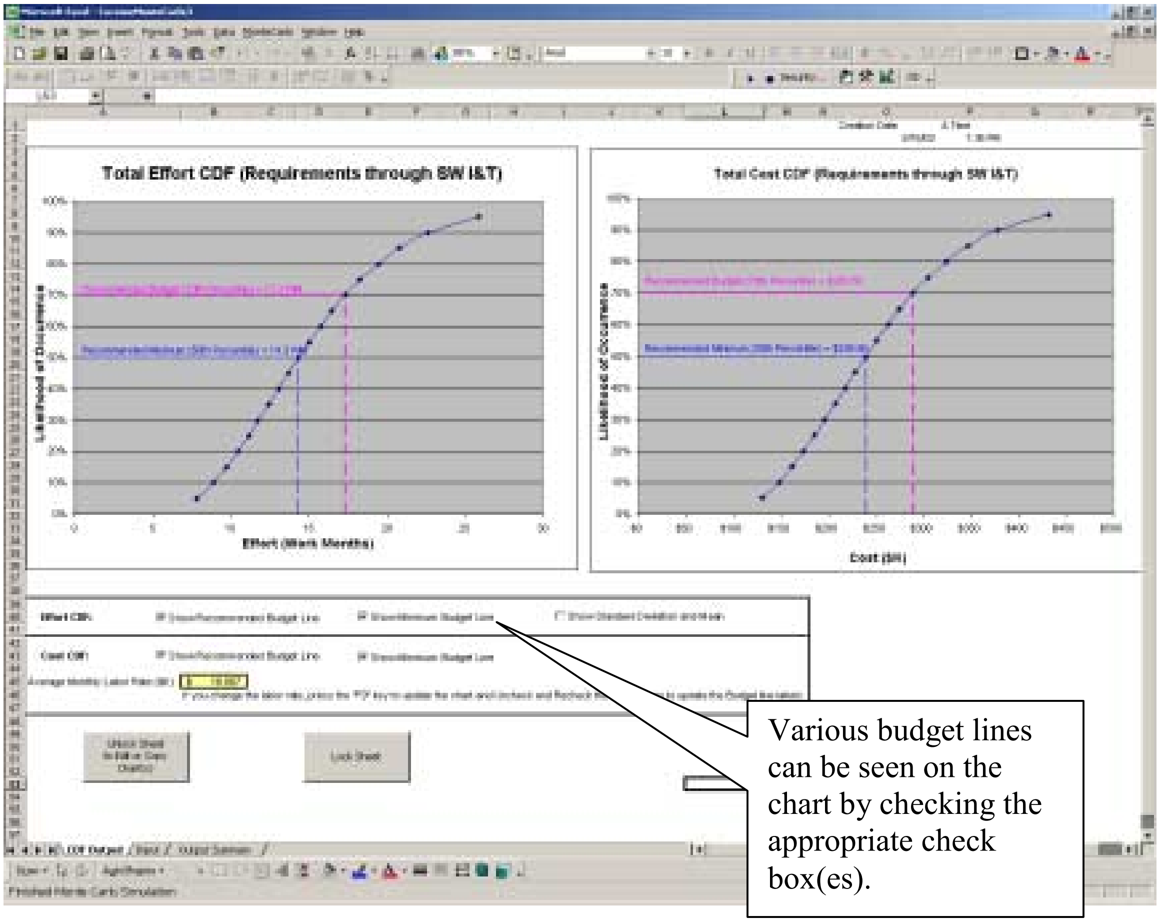
The x-axis is cost in thousands of dollars; the y-axis is probability or likelihood of occurrence.

The cost CDF is based on the average monthly labor rate times the effort estimate at each probability level.

Step **Action**
(1) Enter the average monthly labor rate in thousands of dollars in the yellow cell.
(2) Press the “F9” key to update the curve on the Cost CDF chart
(3) Uncheck and recheck the check boxes, for the budget line(s) you wish to see. This updates the labels for the budget lines.

Options and features: You may view your CDF charts with or without the lines. Use the check boxes to turn the lines on or off. The following lines can be viewable by selecting the check boxes:

- 70th percentile is the recommended budget with reserves
- 50th percentile is the recommended minimum budget without reserves



Various budget lines can be seen on the chart by checking the appropriate check box(es).

Figure 3. CDF Output Sheet

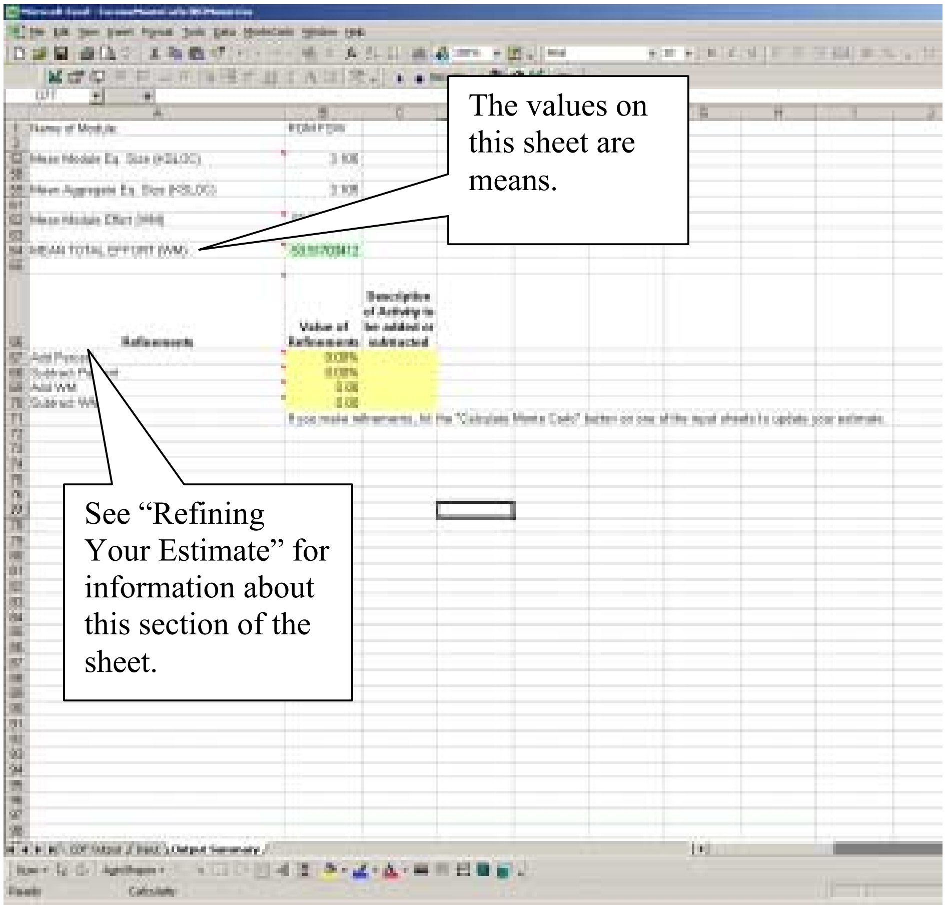
4.2.2 Reading the Output Summary Sheet

The "Output Summary" sheet shows the mean values for size of module, total project size, module effort, and total project effort. The values on this sheet are mean point estimates. The columns list the modules in your project. The rows are as follows:

a. *Mean Module Eq. Size (KSLOC)* – the mean equivalent size in thousands of source lines of code per module.
b. *Mean Aggregate Eq. Size (KSLOC)* – the mean total equivalent size in thousands of source lines of code for the project.
c. *Mean Module Effort (WM)* – the mean effort estimate per module in work-months; this effort estimate includes requirements phase[4] through software I&T.
d. *MEAN TOTAL EFFORT (WM)* – the mean total effort estimate of the project in work-months; this effort estimate includes requirements phase through software I&T.

See "What's Included in the SCAT Estimate, What's Not" for detail of what activities and phases are included in the SCAT effort and cost estimate.

[4] Requirements phase is not usually included in the USC-based version of the COCOMO II tool. However, it is an add-on phase that can show up in one of its reports. We have included the requirements phase so that the estimate is a full software life-cycle estimate.



Name of Module:	[illegible]
Mean Module Eq. Size (KSLOC)	[illegible]
Mean Aggregate Eq. Size (KSLOC)	[illegible]
Mean Module Effort (WM)	[illegible]
MEAN TOTAL EFFORT (WM)	[illegible]

Refinements	Value of Refinements	Description of Activity to be added or subtracted
Add Percent	0.00%	
Subtract Percent	0.00%	
Add WM	0.00	
Subtract WM	0.00	

If you make refinements, hit the "Calculate Monte Carlo" button on one of the input sheets to update your estimate.

The values on this sheet are means.

See “Refining Your Estimate” for information about this section of the sheet.

Figure 4. Output Summary Sheet

4.3 Starting an Estimate

Before starting any estimate, save the file under a new name.

<u>Step</u>	<u>Action</u>
(1)	From the menu bar, select “File”
(2)	Select “Save As”
(3)	Rename the file. Tip: Choose a name that describes the project you are estimating and add the date to the end of the file name so that you can track your changes more easily.
(4)	Select “Save”

Begin selecting inputs (See Appendix A for help) for each module on the input sheet (See “Input Area” for more information). Create additional input sheets for each of your modules (as needed) by selecting the “Add New Module” button.

4.4 Running the Monte Carlo Simulation

Once all of your inputs have been selected for all the modules of the project, click the “Calculate Monte Carlo” button. This will begin a 9,999 iteration Monte Carlo simulation of the possible effort estimates. This may take anywhere from a few seconds to many minutes, depending on the number of modules or input sheets that you have created.

While the simulation is running, you will probably not be able to access other applications on your computer or they may run slower. In addition, other hidden sheets that are necessary for the computation of the effort will be temporarily unhidden. When the Monte Carlo simulation is finished, a sheet named “CDF output” will be created. See “Reading the CDF Output Sheet” for more information.

Make sure to save your estimate when finished.

Step **Action**

(1) Hit the “Calculate Monte Carlo” button.

(2) When the simulation has completed, save the file. It is recommended that you save your estimate under a new file name, if you have not already done so. From the menu bar, select “File,” then “Save As,” and then rename the file.

4.5 What’s Included in the SCAT Estimate, What’s Not

The following phases are included in the SCAT estimate:

- Software Requirements Analysis
- Software Architectural Design
- Software Detail Design
- Implementation (Coding/Programming, Unit Test)
- Software Integration and Test (SW I&T)

The following activities are included in the SCAT estimate:

- Software management
- Software system engineering
- Informal Configuration Management
- Informal software quality assurance
- Documentation

The following phases are *not* included in the SCAT estimate:

- Concept phase
- Maintenance
- System Requirements Analysis
- Hardware/Software Integration and Test
- Acceptance Test
- System Test

The following activities are *not* included in the SCAT estimate:

- Independent Verification and Validation

- Assembly, test, & launch operations
- System-level test support
- Hardware management
- Proposal-level requirements
- Computer center operators
- Personnel-department personnel
- System-level engineering functions
- Secretaries
- Formal Software Quality Assurance
- Higher-level program management

To include or exclude phases or activities that are not in the original SCAT estimate, see “Refining Your Estimate” in “Advanced Features.”

4.6 Changing Your Inputs to Analyze the Tradeoffs

You may wish to explore and compare the impact of various scenarios by changing some of the inputs. In this case, you may switch between input sheets using the tabs on the lower left corner of the Excel sheet and make the appropriate changes. When you are ready, click the “Calculate Monte Carlo” button and an updated “CDF output” sheet will be created. Save this file under a different name to help keep track of your various estimates.

5.0 **ADVANCED FEATURES**

The SCAT workbook has been locked to prevent the links from accidentally being broken. However, there may be situations in which you may need to modify the estimate or change the chart.

5.1 Modifying the CDF Chart

In order to modify the CDF Chart, you need to unlock the sheet. To unlock the "CDF Output" Sheet:

<u>Step</u>	<u>Action</u>
(1)	Press the "Unlock Sheet to Edit or Copy Chart(s)" button
(2)	You can change now the title of chart, change the names of the axes, add lines, format the font type, modify the colors and patterns of the chart.
(3)	When you are done, press the "Lock Sheet" button.

5.2 Copying the CDF Chart to Another Media

You can copy the chart to another Microsoft Office application, such as Microsoft Word, or Microsoft PowerPoint. However, to do this, you must first unlock the "CDF Output" sheet. Follow the instructions in "Modifying the CDF Chart" to unlock the "CDF Output" sheet. Then,

<u>Step</u>	<u>Action</u>
(1)	Click on the CDF chart so that it is selected
(2)	Select "Edit" on your Excel menu bar
(3)	Select "Copy"
(4)	Open the Word document or PowerPoint file in which you wish to paste the chart.
(5)	Move your cursor to where you wish to paste the chart
(6)	Select "Edit" on the Word or PowerPoint menu bar
(7)	Select "Paste Special"
(8)	Select the "Picture (Enhanced Metafile)" option. This is the recommended format in which to paste a chart. Pasting the chart as an Excel Object takes up a large amount of disk space and can slow down your applications.
(9)	Click "OK"
(10)	Format the picture to your liking.
(11)	Return to your "CDF Output" sheet and press the "Lock Sheet" button.

5.3 Refining your Estimate

There may be situations in which you need to refine your estimate. For example, you may not want to include the requirements phase in your estimate, or you may need to shift the CDF curve for activities or items that were not included in the tool's estimates.

To refine your estimate:

<u>Step</u>	<u>Action</u>
(1)	Save your original estimate.
(2)	Save this refined estimate using a different file name.
(3)	Select the "Output Summary" Sheet

(4) Enter the refinements you wish to make in the first column of yellow-shaded cells. Enter a description of the activity that you are adding or subtracting from the estimate in order to refine the estimate. The following is a description of the fields you can use for your refinements:

a. *Add Percent* – Amount of effort as a percentage of total effort (requirements through software I&T) that you wish to add to the current total estimate

b. *Subtract Percent* – Amount of effort as a percentage of total effort (requirements through software I&T) that you wish to take off the current total estimate

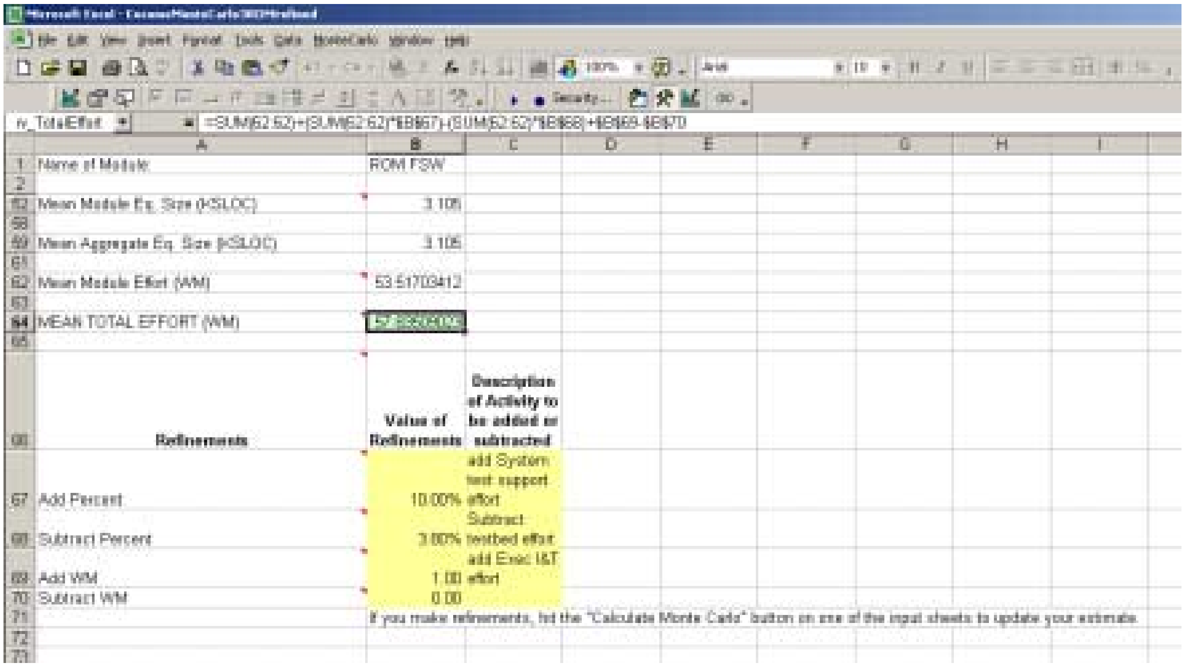
c. *Add WM* – Amount of effort in absolute work-months that you wish to add to the current total estimate

d. *Subtract WM* – Amount of effort in absolute work-months that you wish to subtract from the current total estimate

Software Cost Estimation with COCOMO II by Boehm, et al. is a helpful reference in determining the percentage of effort or amount of effort you should add or subtract for the activities you wish to include or exclude from your estimate.

(5) Go to one of the input sheets and press the “Calculate Monte Carlo” button.

(6) The CDF curves on your “CDF Output” sheet will shift.


File Edit View Insert Format Tools Data MonteCarlo Window Help
100%
Arial
10
Security...

rv_TotalEffort

=SUM(62:62)+(SUM(62:62)*B67)-(SUM(62:62)*B68)+B69-B70

	A	B	C
1	Name of Module	ROM FSW	
2			
52	Mean Module Eq. Size (KSLOC)	3.105	
58			
59	Mean Aggregate Eq. Size (KSLOC)	3.105	
61			
62	Mean Module Effort (WM)	53.51703412	
63			
64	MEAN TOTAL EFFORT (WM)	[illegible]	
65			
66	Refinements	Value of Refinements	Description of Activity to be added or subtracted
67	Add Percent	10.00%	add System test support effort
68	Subtract Percent	3.00%	Subtract testbed effort
69	Add WM	1.00	add Enac I&T effort
70	Subtract WM	0.00	
71		If you make refinements, hit the "Calculate Monte Carlo" button on one of the input sheets to update your estimate.	
72			
73			

Figure 5. Refinements are made on the “Output Summary” Sheet

5.4 Tips and Tricks

- ○ Save your estimate with a new file name each time you make modifications.

- If you create too many input sheets, do not attempt to delete the sheets. Just set all the size inputs to '0' so that the inputs on this sheet have no effect.
- When pasting your charts to other applications, paste them as a picture. Pasting the chart as an Excel Object takes up a large amount of disk space and can slow down your applications.
- When saving your estimates, choose a file name that describes the project you are estimating (such as an acronym of the project name) and add the date to the end of the file name so that if you perform other estimates for the same project you can compare and track your estimates easily.
- If you make any changes to the parameters and rerun the Monte Carlo simulation, it is a good idea to save the estimate under a new file name for configuration management purposes.
- Record your basis of estimates in the yellow-shaded cells of the fifth column of the input sheets. This can be very helpful in tracking and understanding the changes you made while developing the estimate.

5.5 Error Detection

While SCAT has been designed with flexibility and extensibility in mind, it is vulnerable to the occasional input error. However, SCAT will do its best to programmatically catch these errors at run time (Calculate Estimate). When an error is caught, READ the message box that pops up. In most cases, the message box will tell you how to correct the problem.

Some errors may slip past the error detection routines. If this occurs, please contact the tool developer for a prompt resolution to your problem.

APPENDIX A. COCOMO II PARAMETERS AND RATING SCALE

Table 1. COCOMO II Parameters and Rating Scale

CATEGORY/ Parameters	**Recommendations/Rating Scale**
LINES OF CODE	
Size	Enter your size estimates from Software Estimation Step #3 for each low-level element. Or if using analogy to historical data based on physical SLOC, convert physical SLOC to logical SLOC. In general, estimators tend to be overly optimistic on the amount of code that can be inherited from projects. Therefore, it is better to underestimate the size of inherited/reused software.
% Design Modified	If there is heritage, enter % of inherited design to be modified.
% Code Modified	If there is heritage, enter % of the inherited or reused code that will be modified.
% Integration Modified	If there is heritage, enter % of the effort needed for integrating and testing the adapted software as compared to the normal amount of integration and test effort for software of comparable size.
% Code breakage	Enter % of code thrown away due to requirements evolution and volatility.

Post Architecture Effort Multipliers	**Very Low**	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
PRODUCT ATTRIBUTES						
RELY Required Software Reliability	Effect of SW failure = slight inconvenience (0.82)	Effect of SW failure = low, easily recoverable losses (0.92)	Effect of SW failure = moderate, easily recoverable losses (1.00)	Effect of SW failure = high financial loss (1.10)	Effect of SW failure = risk to human life/public safety requirements (1.26)	
DATA Database Development Size		Testing DB Bytes/Program SLOC < 10 (0.90)	10 ≤ D/P < 100 (1.00)	100 ≤ D/P < 1000 (1.14)	D/P ≥ 1000 (1.28)	
CPLX Product Complexity	See Table 2					
DOCU Documentation Match to Life-Cycle Needs	Many life-cycle needs uncovered (0.81)	Some life-cycle needs uncovered (0.91)	Right-sized to life-cycle needs (1.00)	Excessive for life-cycle needs (1.11)	Very excessive for life-cycle needs (1.23)	
RUSE Developed for Reusability		None (0.95)	Across project (1.00)	Across program (1.07)	Across product line (1.15)	Across multiple product lines (1.24)
PLATFORM ATTRIBUTES						
TIME Execution Time Constraint			≤50% use of available execution time (1.00)	70% use of available execution time (1.11)	85% use of available execution time (1.29)	95% use of available execution time (1.63)
STOR Main Storage Constraint			≤50% use of available storage (1.00)	70% use of available storage (1.05)	85% use of available storage (1.17)	95% use of available storage (1.46)
PVOL Platform Volatility		Major change every 12 mo.; Minor change every 1 mo. (0.87)	Major change every 6 mo.; Minor change every 2 wk. (1.00)	Major change every 2 mo.; Minor change every 1 wk. (1.15)	Major change every 2 wk.; Minor change every 2 days (1.30)	
PERSONNEL ATTRIBUTES The personnel attributes are the most misused of the all the effort multipliers. If you do not know who you will be hiring, then assume Nominal, which would represent average capability and experience.						
ACAP Analyst Capability	15th percentile (1.42)	35th percentile (1.19)	55th percentile (1.00)	75th percentile (0.85)	90th percentile (0.71)	
PCAP Programmer Capability	15th percentile (1.34)	35th percentile (1.15)	55th percentile (1.00)	75th percentile (0.88)	90th percentile (0.76)	
PCON Personnel Continuity	Annual personnel turnover: 48%/year (1.29)	24%/year (1.12)	12%/year (1.00)	6%/year (0.90)	3%/year (0.81)	

APEX Applications Experience	≤2 months (1.22)	6 months (1.10)	1 year (1.00)	3 years (0.88)	6 years (0.81)	
PLEX Platform Experience	≤2 months (1.19)	6 months (1.09)	1 year (1.00)	3 years (0.91)	6 years (0.85)	
LTEX Language and Tool Experience	≤2 months (1.20)	6 months (1.09)	1 year (1.00)	3 years (0.91)	6 years (0.84)	
PROJECT ATTRIBUTES						
TOOL Use of Software Tools	Edit, code, debug (1.17)	Simple, frontend, backend, CASE, little integration (1.09)	Basic life-cycle tools, moderately integrated (1.00)	Strong, mature life-cycle tools, moderately integrated (0.90)	Strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse (0.78)	
SITE Multisite Development	Collocation: international; Communications: some phone, mail (1.22)	Collocation: multicity and multicompany; Communications: individual phone, fax (1.09)	Collocation: multicity or multicompany; Communications: narrow band email (1.00)	Collocation: same city or metro area; Communications: wideband electronic communication (0.96)	Collocation: same building or complex; Communications: wideband electronic communication, occasional video conf. (0.86)	Collocation: Fully collocated; Communications: Interactive multimedia (0.80)
SCED Required Development Schedule	75% of nominal (1.43)	85% of nominal (1.14)	100% of nominal (1.00)	130% of nominal (1.00)	160% of nominal (1.00)	
SCALE FACTORS	**Very Low**	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
PREC Precedentedness	thoroughly unprecedented (6.20)	largely unprecedented (4.96)	Somewhat unprecedented (3.72)	generally familiar (2.48)	largely familiar (1.24)	thoroughly familiar (0.00)
FLEX Development Flexibility	Rigorous (5.07)	occasional relaxation (4.05)	Some relaxation (3.04)	General conformity (2.03)	Some conformity (1.01)	general goals (0.00)
RESL Architecture/Risk Resolution	little (20%) (7.07)	some (40%) (5.65)	often (60%) (4.24)	Generally (75%) (2.83)	mostly (90%) (1.41)	full (100%) (0.00)
TEAM	very difficult interactions (5.48)	some difficult interactions (4.38)	Basically cooperative interactions (3.29)	Largely cooperative (2.19)	Highly cooperative (1.10)	Seamless interactions (0.00)
PMAT Process Maturity	CMM Level 1 (Lower half) (7.80)	CMM Level 1 (Upper half) (6.24)	CMM Level 2 (4.68)	CMM Level 3 (3.12)	CMM Level 4 (1.56)	CMM Level 5 (0.00)

Table 2. COCOMO II Complexity Table

	Control Operations	Computational Operations	Device-dependent Operations	Data Management Operations	User Interface Management Operations
Very Low (0.73)	Straight-line code with a few non-nested structured programming operators: DOs, CASEs, IF-THEN-ELSEs. Simple module composition via procedure calls or simple scripts.	Evaluation of simple expressions: e.g., A = B + C * (D –E)	Simple read, write statements with simple formats.	Simple arrays in main memory. Simple COTS-DB queries, updates.	Simple input forms, report generators.
Low (0.87)	Straightforward nesting of structured programming operators. Mostly simple predicates	Evaluation of moderately level expressions: e.g., D = SQRT (B*2-4. * A * C)	No cognizance needed of particular processor or I/O device characteristics. I/O done at GET/PUT level.	Single file subsetting with no data structure changes, no edits, no intermediate files. Moderately complex COTS-DB queries, updates.	Use of simple graphic user interface (GUI) builders.
Nominal (1.00)	Mostly simple nesting. Some inter-module control. Decision tables. Simple callbacks or message passing, including middleware-supported distributed processing	Use of standard math and statistical routines. Basic matrix/vector operations.	I/O processing includes device selection, status checking, and error processing.	Multi-file input and single file output. Simple structural changes, simple edits. Complex COTS-DB queries, updates.	Simple use of widget set.
High (1.17)	Highly nested structured programming operators with many compound predicates. Queue and stack control. Homogeneous, distributed processing. Single processor soft real-time control.	Basic numerical analysis: multivariate interpolation, ordinary differential equations. Basic truncation, round off concerns.	Operations at physical I/O level (physical storage address translations; seeks, reads, etc.). Optimized I/O overlaps.	Simple triggers activated by data stream contents. Complex data restructuring.	Widget set development and extension. Simple voice I/O multimedia.
Very High (1.34)	Reentrant and recursive coding. Fixed-priority interrupt handling. Tasks synchronization, complex callbacks, heterogeneous distributed processing. Single-processor hard real-time control.	Difficult but structured numerical analysis: near-singular matrix equations, partial differential equations. Simple parallelization.	Routines for interrupt diagnosis, servicing, masking. Communication line handling. Performance-intensive embedded systems.	Distributed database coordination. Complex triggers. Search optimization.	Moderately complex 2D/3D, dynamic graphics, multimedia.
Extra High (1.74)	Multiple resource scheduling with dynamically changing priorities. Microcode-level control. Distributed hard real-time control.	Difficult and unstructured numerical analysis: highly accurate analysis of noisy, stochastic data. Complex parallelization.	Device timing-dependent coding, micro-programmed operations. Performance-critical embedded systems.	Highly coupled, dynamic relational and object structures. Natural language data management.	Complex multimedia, virtual reality.

APPENDIX B. GLOSSARY

Term	Description
Adapted SLOC	Amount of reused code
ATLO	Assembly, Test & Launch Operations. Formal test performed for all software included in a flight mission.
Auto Generated	Delivered code that is produced by a code translator that transforms code developed in a script language or graphical code development tool into a standard language such as C or C++.
Effort	The time it takes one or more persons to accomplish an activity. Activities can include software management, software system engineering, software engineering, and test engineering. All effort is measured in terms of FTEs (Full Time Equivalents). A work month is 20 staff days (160 hours).
Equivalent Lines of Code	The size of new code plus reused and adapted code that is adjusted to its equivalent in new code. The adjustment is based on the additional effort (redesign, recode, and retest) it takes to modify the code for inclusion in the product.
I&T	Integration and Test
IV&V	Independent Verification and Validation
Logical SLOC	Logical statements may encompass several physical lines and typically include executable statements, declarations, and compiler directives. A logical statement is a single software instruction, having a defined beginning and ending independent of any relationship to the physical lines on which it is recorded or printed. Most detailed software cost models, such as SEER-SEM and COCOMO II, require logical lines of code as the software size input.
New	Code written from scratch. New code written manually for the system.
Project Name	The official identifier for the project. Usually an acronym
SLOC	**Source Lines of Code -** a defined standard for the counting of computer instructions and data definitions expressed in a programming language. The count includes executable statements, reuse code and auto generated code but does not include comments, blank lines, Commercial-Off-The-Shelf (COTS) software or Government-Off-The-Shelf (GOTS) software. There are various ways of counting LOC. Most cost models use logical LOC as inputs. The LOC presented in the models in this handbook have been converted from physical LOC to logical LOC.
Software	The set of software, that includes computer programs, procedures, and associated documentation and data pertaining to the operation of flight, mission ground support, or general project support software systems. Computer programs include any and all executable programs whether they are JPL or contractor developed, COTS and GOTS, reuse code, glueware, middleware, firmware, development and test environments, and security measures. Examples of included software: • flight software for on-board spacecraft, aircraft applications, robotics applications, all ground systems in direct support of space and aircraft missions (e.g. fight dynamics, control center, command processing, crew and controller training) • general support of flight and ground software that is tracked and managed as part of a project (e.g. engineering models, simulations, engineering analyses, prototypes, wind tunnel analyses, test aids, science data processing, and tools).
Software Assurance	Activities that perform systematic software evaluation of products and processes to provide adequate confidence that an item, product, or process conforms to established procedures and technical requirements.
Software Development	The period of time that begins with the decision to develop a software product and ends when the software is delivered to operations. This cycle typically includes a requirements phase, design phase, implementation phase, test phase, and sometimes, installation and checkout phase. These phases may overlap or be performed iteratively, depending upon the software development approach. This term is sometimes used to mean a longer period of time, either the period that ends when the software is no longer being enhanced by the developer, or the entire software life cycle. New application, new spacecraft, pre-first article, never been flown before this version or the requirements are changed extensively from existing software. Software Development projects may include some maintenance work.
Software Engineering	Activities performed by the cognizant engineer and developers to unit design, develop code, unit test, and integrate software components.

Term	Description
Software Formulation	Includes the following activities: Architectural design, software requirements, software design, and implementation planning
Software Implementation	Includes the following activities: Detail design, code, unit testing, and software integration and test phases.
Software Integration & Test	Testing the different software component as a unit.
Software Maintenance	Activities for software repair, enhancement, or adaptation performed after development completion. The process of modifying a software system or component after operational readiness to correct faults, improve performance or other attributes, or adapt to a changed environment. Include: Changes to existing software for minimal new functionality, updates for compatibility, and error fixes. Excludes: Changes to existing software for increasing or modifying functionality to meet requirements for a new project. These types of changes should fall under Software Development.
Software Management	Activities performed by the project element manager (PEM), flight software manager, technical lead, and system administration to plan and direct the software project and software configuration management.
Software System Engineering	Includes work from different levels of SW engineering. System Architect, Project Software System Engineer, Subsystem Engineer (SSE), Service System Development Engineer (SSDE) for functional design, software requirements, and interface specification.
Software Test Engineering	Activities performed by a group separate from those involved in software engineering to write test plans and procedures to perform any level of test above unit testing. Does not include test-bed development and support, system-level test support, or ATLO support.
SW	Software
System Test	This includes Integration & Test of S/W with Hardware, ATLO, Acceptance Test, System Test
Work-Month	The amount of time one person spends working on the software project for one month; 1 work-month is usually equivalent to 20 staff days or 160 hours. Example: Two people working full-time for 2 years (24 calendar months) is equal to 48 work-months.

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.